

Санкт-Петербургский государственный университет
Кафедра теории управления

Соколов Владимир Николаевич

Магистерская диссертация

**Упрощенный вариант метода анализа иерархий в
условиях нечетких данных**

Направление 01.04.02

Прикладная математика и информатика

Магистерская программа Методы прикладной математики и
информатики и задачах управления

Научный руководитель,
доктор физ.-мат. наук,
профессор
Ногин В. Д.

Санкт-Петербург
2016

ОГЛАВЛЕНИЕ

	Стр.
АННОТАЦИЯ	4
ВВЕДЕНИЕ	4
ОБЗОР ЛИТЕРАТУРЫ	6
 ГЛАВА 1 Метод анализа иерархий	 8
1.1 Описание подхода	8
1.1.1 Построение иерархий	8
1.1.2 Расстановка приоритетов	10
1.2 Матрица относительных весов	11
1.3 Анализ иерархии	12
1.4 Линейная свертка критериев	14
 ГЛАВА 2 Упрощенный метод анализа иерархий	 15
2.1 Требования к матрице парных сравнений	15
2.2 Построение матрицы парных сравнений	16
 ГЛАВА 3 Теория нечетких множеств	 20
3.1 Основные понятия и определения	20
3.2 Нечеткие числа LR типа	21
3.3 Примеры LR чисел	22
3.4 Операции с нечеткими числами и интервалами	23
3.5 Дефаззификация	25
 ГЛАВА 4 Упрощенный вариант метода анализа иерархий в условиях нечетких данных	 27
4.1 Экспертные оценки	27
4.2 Построение вектора весов	29
4.3 Дефаззификация	31
 ГЛАВА 5 Анализ результатов	 32
5.1 Подготовка данных	32
5.2 Проблема водного хозяйства в Китае	32
5.3 Результаты работы различных методов	34
5.4 Сравнение и анализ методов	37
ЗАКЛЮЧЕНИЕ	41
СПИСОК ЛИТЕРАТУРЫ	42

ПРИЛОЖЕНИЕ	44
-------------------------	-----------

АННОТАЦИЯ

Данная работа посвящена распространению упрощенного варианта метода анализа иерархий (МАИ) на случай нечетких исходных данных. Показывается, как на основе $n-1$ экспертной оценки можно построить нечеткий вектор приоритетов и провести его дефаззификацию. В работе проводится сравнение предложенного метода с близкими к нему подходами и показывается его практическая применимость. Для изучения предлагаемого подхода, выполнена его реализация на языке программирования C# и осуществлено тестирование на задаче поиска распределения инвестиций в развитие водных бассейнов Китая. Работа содержит введение, обзор литературы, пять глав, заключение, список литературы из 15 наименований и код программы.

ВВЕДЕНИЕ

Метод Анализа Иерархий (Далее МАИ) — математический инструмент системного подхода к сложным проблемам принятия решений. МАИ не предписывает лицу, принимающему решение (ЛПР), какого-либо «правильного» решения, а позволяет ему в интерактивном режиме найти такой вариант (альтернативу), который наилучшим образом согласуется с его пониманием сути проблемы и требованиями к её решению. Этот метод разработан американским математиком Томасом Саати [1], который написал о нем книги, разработал программные продукты и в течение 20 лет проводит симпозиумы ISANP (англ. International Symposium on Analytic Hierarchy Process). МАИ широко используется на практике и активно развивается учеными всего мира. В его основе наряду с математикой заложены и психологические аспекты. МАИ позволяет понятным и рациональным образом структурировать сложную проблему принятия решений в виде иерархии, сравнить и выполнить ко-

личественную оценку альтернативных вариантов решения. Метод Анализа Иерархий используется во всем мире для принятия решений в разнообразных ситуациях: от управления на межгосударственном уровне до решения отраслевых и частных проблем в бизнесе, промышленности, здравоохранении и образовании.

Модификация этого метода, называемая Упрощенным МАИ, разработанная Ногиним В.Д. в [2], позволяет получить схожие результаты при меньшей вычислительной сложности. «Упрощение» достигается за счет снижения количества попарных сравнений в обычном МАИ с сохранением лишь минимального набора информации, достаточного для решения поставленной задачи. Стоит отметить, что предлагаемый в этом подходе способ упрощения не единственен, и может быть выбран таким, чтобы максимально соответствовать действительности.

Теория нечетких множеств, разработанная Л. Заде [3], позволяет представлять любую информацию в виде множества с некоторой функцией принадлежности. Эта теория в текущий момент активно развивается и используется в различных сферах математики и прикладных наук. С помощью инструментария, используемого в этой теории, мы рассмотрим известную задачу анализа иерархий в условиях нечетких данных. Такой подход позволит снизить влияние субъективности начальных данных на конечный результат, делая его более объективным и соответствующим реальности.

В первой главе будет рассмотрен и описан классический МАИ, будут введены основные определения и тезисы. Большая часть этой главы будет опираться на результаты полученные в [1].

Во второй главе будут показаны основные этапы упрощенного метода анализа иерархий. В дополнение к этому, будут доказаны некоторые утверждения позволяющие упростить дальнейшую компьютерную реализацию.

В третьей главе изложены основные принципы и понятия требуемые для корректного использования теории нечетких множеств в рамках метода анализа иерархий. Будут введены понятия LR чисел (интервалов) и арифметических операций с ними.

Четвертая глава содержит описание упрощенного метода анализа иерархий в условиях нечетких данных. Данный метод основан на объединении результатов предыдущих глав, и позволяет решить стандартную задачу анализа иерархий за счет меньшего количества попарных сравнений и с учетом «нечеткости» экспертных оценок.

В пятой главе будут показаны результаты компьютерной реализации метода, предлагаемого в данной работе, и произведено его сравнение с основными существующими модификациями МАИ. Сравнение будет проводиться на основании реальных данных полученных из современных источников.

ОБЗОР ЛИТЕРАТУРЫ

В [1], [5] изложены основные результаты теории принятия решений применительно к методу анализа иерархий. Работа [1] является главной отправной точкой для предлагаемого в нашем исследовании метода, и содержит основные используемые определения, леммы и теоремы.

В [2] предложена модификация классического МАИ, предложенная Ногиным Д.В., существенно упрощающая выполняемые вычисления, и требующая меньшего количества экспертных оценок.

В работах [3], [4] представлены основы теории нечетких множеств. Конкретно в работе [3] было впервые предложено использование «нечетких» множеств как альтернативы детерминированным числам и множествам.

Работы [6]-[8], [11], [12] содержат информацию, позволяющую сузить рассматриваемый спектр нечетких множеств, и формализовать арифметические операции над ними. В дополнение к этому, на результатах, полученных из этих источников, основывается компьютерная реализация работы с нечеткими числами, предложенная в нашей работе.

В работе [9] предложена нечеткая вариация метода анализа иерархий, с применением дискретных нечетких множеств. Для построения

вектора весов, задача поиска приоритетов сводится к задаче линейного программирования, что является довольно таки нестандартным подходом к решению такого рода задач. Кроме того, данная работа будет использована для сравнения с предлагаемым нами подходом на основе имеющихся в ней данных.

В [10] представлен еще один способ решения задачи анализа иерархий, основывающийся на теории нечетких множеств. В этой работе задача поиска вектора приоритетов сведена к решению задачи нелинейной оптимизации. Также сравнены и проанализированы три различных подхода поиска требуемых экстремумов.

В работе [13] представлена модификация МАИ, также базирующаяся на теории нечетких множеств. В качестве экспертных оценок используются нечеткие числа треугольного типа. Эта работа представляет собой один из классических примеров применения нечеткой логики к задаче анализа иерархий.

В [14] содержится пример, на основе которого будет производиться сравнение рассматриваемых методов.

В работе [15] содержатся сведения по теории графов, требуемые для доказательства некоторых утверждений.

ГЛАВА 1

Метод анализа иерархий

1.1 Описание подхода

Метод анализа иерархий содержит процедуру синтеза приоритетов, вычисляемых на основе субъективных суждений экспертов. Число суждений может измеряться дюжинами или даже сотнями. Математические вычисления для задач небольшой размерности можно выполнить вручную или с помощью калькулятора, однако гораздо удобнее использовать программное обеспечение (ПО) для ввода и обработки суждений. Самый простой способ компьютерной поддержки — электронные таблицы. Самое развитое ПО предусматривает применение специальных устройств для ввода суждений участниками процесса коллективного выбора. Порядок применения Метода Анализа Иерархий:

1. Построение качественной модели проблемы в виде иерархии, включающей цель, альтернативные варианты достижения цели и критерии для оценки качества альтернатив.
2. Определение приоритетов всех элементов иерархии с использованием метода парных сравнений.
3. Синтез глобальных приоритетов альтернатив путем линейной свертки приоритетов элементов на иерархии.
4. Проверка суждений на согласованность.
5. Принятие решения на основе полученных результатов.

Рассмотрим эти этапы подробнее.

1.1.1 Построение иерархий

Первый шаг МАИ — построение иерархической структуры, объединяющей цель выбора, критерии, альтернативы и прочие факторы,

влияющие на выбор решения. Построение такой структуры позволяет проанализировать все аспекты имеющейся проблемы и глубже вникнуть в суть задачи.

Иерархическая структура — это графическое представление задачи в виде дерева, где каждый элемент, за исключением самого верхнего, зависит от одного или более элементов расположенных выше. Похожие структуры зачастую встречаются в жизни (структура компании, распределение полномочий, таксономическая классификация и т.д.)

Иерархические структуры используются для лучшего понимания сложной реальности: мы раскладываем исследуемую проблему на составные части; затем разбиваем на составные части получившиеся элементы и т. д. На каждом шаге важно фокусировать внимание на понимании текущего элемента, временно абстрагируясь от всех прочих компонентов. При проведении подобного анализа приходит понимание всей сложности и многогранности исследуемого предмета. Иерархические структуры, ис-

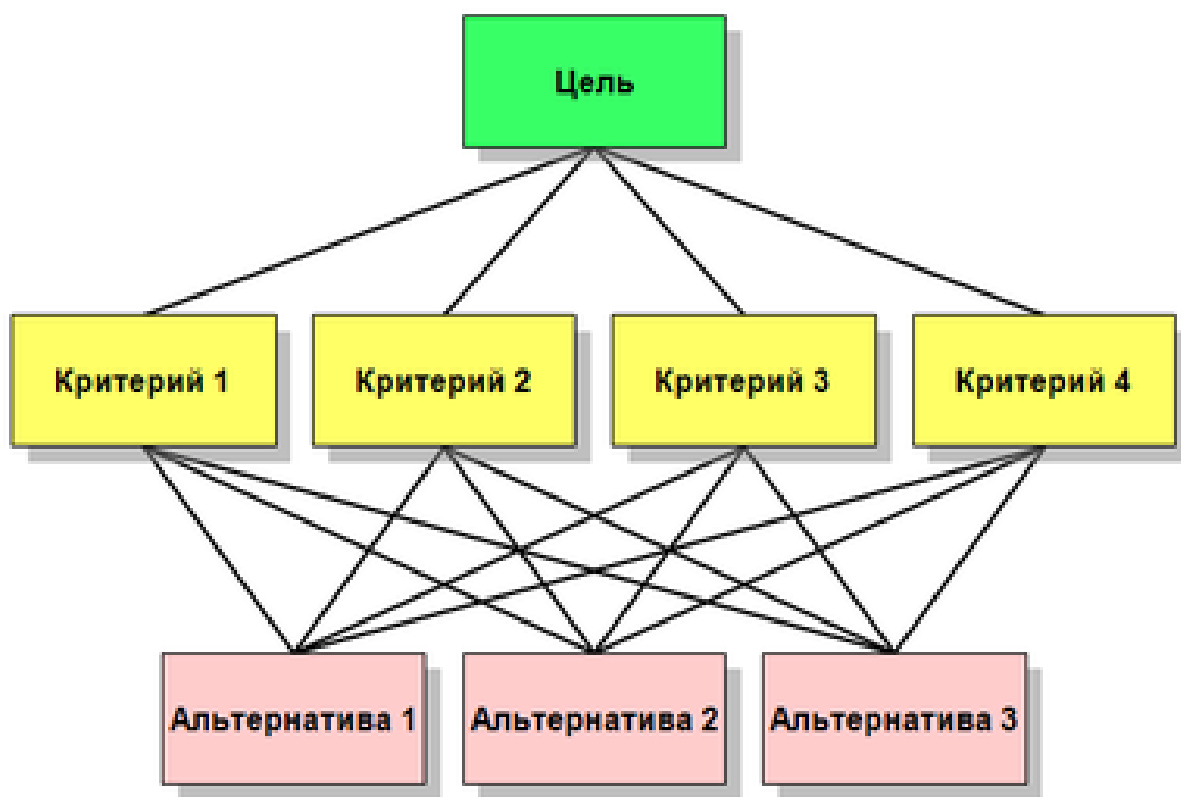


Рис. 1.1: Простейшая иерархия МАИ.

пользуемые в МАИ, представляют собой инструмент предназначенный

для моделирования сложных проблем. На вершине иерархии находится главная цель; элементы самого нижнего уровня представляют собой множество вариантов достижения цели (альтернатив); остальные элементы промежуточных уровней соответствуют критериям или факторам, которые связывают конечную цель с альтернативами. (см. Рис. 1.1)

Также существует другой способ построения иерархии. Он заключается в расчленении рассматриваемых вещей на большие группы или кластеры, которые далее делятся на меньшие кластеры и т.д. Тогда целью будет получение приоритетов всех элементов посредством группирования. Это также намного более эффективный процесс, чем обработка всех элементов совместно.

В [1] показано, что желательная размерность одного кластера равна 7 ± 2 элементам. Такое разбиение позволяет достичь оптимального оценивания альтернатив, уменьшая при этом количество попарных сравнений.

Следовательно, несущественно, думаем ли мы, что иерархии внутренне присущи природе (как утверждают некоторые исследователи), или же мы просто используем их из-за ограниченной способности обрабатывать информацию. В любом случае они представляют собой очень эффективный способ исследования сложных проблем.

1.1.2 Расстановка приоритетов

Приоритеты — это числа, связанные с узлами иерархии, представляющие собой относительные веса элементов в каждой группе. Подобно вероятностям, приоритеты — это безразмерные величины, принимающие значения от нуля до единицы. Чем больше величина приоритета, тем более значимым является соответствующий ему элемент. Сумма приоритетов элементов, которые подчинены одному элементу выше лежащего уровня иерархии, равна единице. Приоритет цели по определению также равен 1.

На рисунке 1.2 показан пример иерархии с вычисленными значениями веса критериев по отношению к цели, и вектора весов каждой из

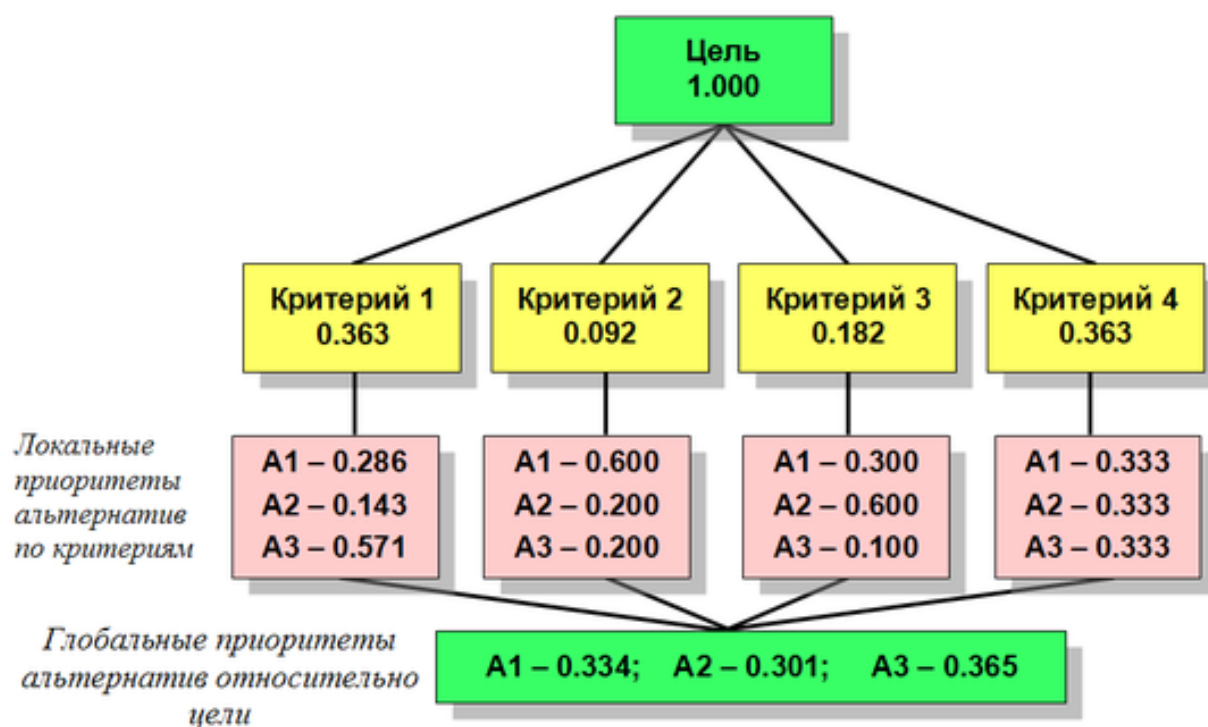


Рис. 1.2: Иерархическая структура с некоторыми значениями приоритетов

альтернатив по данным критериям. Путем линейной свертки получены конечные веса альтернатив по исследуемому целевому критерию. Подробнее процесс получения глобальных и локальных приоритетов будет рассмотрен далее в этой работе.

1.2 Матрица относительных весов

В методе анализа иерархий активно используется понятие матрицы относительных весов. Введем формальное определение такой матрицы.

Пусть задан набор из N элементов, обозначаемых $A_1, A_2, A_3, \dots, A_n$. Предположим, что каждому из элементов A_k поставлено в соответствие некоторое положительное число w_k . Будем называть это число весом элемента $A_k, k = 1, 2, \dots, n$.

Матрицу следующего вида:

$$A = (a_{ij})_{n \times n} = \begin{pmatrix} w_1/w_1 & w_1/w_2 & \dots & w_1/w_n \\ w_2/w_1 & w_2/w_2 & \dots & w_2/w_n \\ \vdots & \vdots & \ddots & \vdots \\ w_n/w_1 & w_n/w_2 & \dots & w_n/w_n \end{pmatrix} \quad (1.1)$$

Будем называть матрицей относительных весов.

Для этой матрицы справедливы следующие свойства:

1. Все элементы матрицы A положительны.
2. Матрица A обратно симметрична. То есть $a_{ij} = w_i/w_j = \frac{1}{w_j/w_i} = 1/a_{ji}$
3. Матрица A обладает свойством совместности, а именно: $a_{ik}a_{kj} = \frac{w_i w_k}{w_k w_j} = \frac{w_i}{w_j} = a_{ij}$.
4. Число n является наибольшим собственным числом матрицы A , а вектор $w = (w_1, w_2, \dots, w_n)^T$ единственным (с точностью до нормировки) соответствующим ему правым собственным вектором, то есть:

$$Aw = nw \quad (1.2)$$

При таком определении матрицы A подразумевается, что веса w_1, w_2, \dots, w_n определены заранее. Однако при реализации МАИ, значения весов не известны и подлежат вычислению. Таким образом, задача анализа иерархий ставится следующим образом: для элементов A_1, A_2, \dots, A_n найти вектор w_1, w_2, \dots, w_n (обычно нормированный) их весов по целевому критерию.

1.3 Анализ иерархии

В классическом МАИ на каждом уровне имеющихся иерархий требуется определить вектор приоритетов по критерию стоящему выше в иерархии. Для каждого уровня этот процесс проходит аналогичным образом, суть которого заключается в следующем.

Сначала строится матрица парных сравнений:

$$A = (a_{ij})_{n \times n} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} \quad (1.3)$$

Произвольный элемент a_{ij} представляет собой положительное число, показывающее во сколько раз элемент A_i предпочтительней элемента A_j по рассматриваемому критерию. Данная матрица строится на основе экспертных оценок, являющихся качественной оценкой «предпочтительности» одного элемента над другим. В классических методах анализа иерархий [1] вводится специальная шкала сравнений, называемая шкалой Саати (см. Рис. 1.3), описывающая правило получения значения a_{ij} на основе парного сравнения A_i и A_j по какому-либо критерию.

Степень важности	Определение	Объяснение
1	Одинаковая значимость	Два действия вносят одинаковый вклад в достижение цели
3	Некоторое преобладание значимости одного действия перед другим (слабая значимость)	Опыт и суждение дают лёгкое предпочтение одному действию перед другим
5	Существенная или сильная значимость	Опыт и суждение дают сильное предпочтение одному действию перед другим
7	Очень сильная или очевидная значимость	Предпочтение одного действия перед другим очень сильно. Его превосходство практически явно.
9	Абсолютная значимость	Свидетельство в пользу предпочтения одного действия другому в высшей степени предпочтительны
2, 4, 6, 8	Промежуточные значения между соседними значениями шкалы	Ситуация, когда необходимо компромиссное решение
Обратные величины приведённых выше чисел	Если действию i при сравнении с действием j приписывается одно из приведённых выше чисел, то действию j при сравнении с i приписывается обратное значение	Обоснованное предположение
Рациональные значения	Отношения, возникающие в заданной шкале	Если постулировать согласованность, то для получения матрицы требуется n числовых значений

Рис. 1.3: Качественная шкала Саати

Матрица A инициализируется за счет $\frac{n(n-1)}{2}$ попарных сравнений (остальные элементы находятся на основе свойства обратной симметричности).

Полученная таким образом матрица в большинстве случаев не обладает свойством совместности. Происходит это по причине того, что выставляемые оценки субъективны и выставляются приблизительно, чтобы упростить работу экспертов.

Далее находится λ_{max} максимальное собственное число матрицы 1.3, которое, к слову, всегда $\geq n$. Затем ищется собственный вектор, соответствующий λ_{max} , который и берется в качестве искомого вектора приоритетов.

Основная проблема МАИ заключается в том, что зачастую λ_{max} сильно превосходит размерность матрицы A . Это приводит к несогласованности полученного вектора весов и матрицы парных сравнений, в том смысле, что $a_{ij} \neq \frac{w_i}{w_j}$. Тем самым, получается своего рода «модельная» ошибка МАИ, не позволяющая всегда получать удовлетворительные результаты.

1.4 Линейная свертка критериев

Пусть мы имеем задачу анализа иерархий с более чем одной иерархией в структуре. После получения матрицы парных сравнений и расчета вектора приоритетов по каждому из критериев, необходимо получить вектор приоритетов рассматриваемых альтернатив по целевому критерию.

Пусть у нас есть приоритеты какой-либо альтернативы a по каждому из критериев иерархии, к которым она принадлежит: a_1, a_2, \dots, a_n . В свою очередь, вектор весов рассматриваемых критериев по отношению к какому-либо элементу K^* из вышестоящей иерархии имеет вид: k_1, k_2, \dots, k_n . Тогда вес альтернативы a по критерию K^* будут иметь вид: $w_a = \sum_1^n a_i k_i$. Производя аналогичные вычисления по отношению к каждой имеющейся альтернативе и продвигаясь таким образом вверх по иерархии получаем вектор приоритетов исследуемых альтернатив по целевому критерию.

ГЛАВА 2

Упрощенный метод анализа иерархий

Этот метод предложен в [2] и представляет собой модификацию МАИ, позволяющую найти конечный вектор приоритетов с использованием меньшего количества экспертных оценок. Благодаря применению этого подхода, можно ускорить процесс принятия решения, решив при этом проблему несогласованности матрицы A .

2.1 Требования к матрице парных сравнений

Пусть у нас есть набор из N объектов и надо определить вес каждого из них по какому-либо критерию. Другими словами, требуется определить вектор весов: w_1, w_2, \dots, w_n .

Для начала потребуем от матрицы парных сравнений выполнения свойств (1)-(4), а именно:

1. Все элементы матрицы A положительны: $a_{ij} > 0; \forall i, j = 1, 2, \dots, n$.
2. Матрица A обратно симметрична: $a_{ij} = 1/a_{ji}$.
3. Матрица A обладает свойством совместности: $a_{ik}a_{kj} = a_{ij}$.
4. Число n является наибольшим собственным числом матрицы A , а вектор $w = (w_1, w_2, \dots, w_n)^T$ единственным (с точностью до нормировки) вектором, удовлетворяющим (1.2)

Заметим, что свойство (4) является избыточным, так как справедлива следующая лемма [2]:

Лемма 1 *Любая матрица A удовлетворяющая свойствам (1)-(3), удовлетворяет и свойству (4).*

2.2 Построение матрицы парных сравнений

Пусть в результате экспертного оценивания мы получили элементы первой строки матрицы A . При условии выполнения свойств (1)-(3), имеют место следующие формулы:

$$a_{ij} = a_{i1}a_{1j} = \frac{a_{1j}}{a_{1i}}, i = 2, 3, \dots, n; j = 1, 2, \dots, n \quad (2.1)$$

А значит, используя элементы 1-й строки, мы можем построить всю матрицу парных сравнений, которая будет обладать свойством совместности.

Справедлива следующая теорема [2]:

Теорема 2.2.1 Пусть квадратная матрица A размерности n удовлетворяет свойствам (1)-(2), и построена на основе элементов первой строки в соответствии с формулой (2.1). Тогда справедливы утверждения:

1. Матрица A обладает свойствами (3)-(4) и при этом не существует другой матрицы с такой-же первой строкой, и обладающей свойствами (3)-(4);
2. Максимальное собственное число такой матрицы $\lambda_{max} = n$
3. Собственный вектор w этой матрицы, соответствующий максимальному собственному числу n удовлетворяет формуле: $w_i = \frac{a_{1n}}{a_{1i}}$

Доказательство. Утверждения 1 и 2 доказаны в [1] и [2], покажем справедливость 3 утверждения.

Возьмем вектор $w = (w_1, w_2, \dots, w_n)^T = (\frac{a_{1n}}{a_{11}}, \frac{a_{1n}}{a_{12}}, \dots, \frac{a_{1n}}{a_{1n}})^T$ и подставим его в уравнение $Aw = nw$. После подстановки в левую часть уравнения и небольшого преобразования получаем:

$$Aw = \begin{pmatrix} a_{11} \frac{a_{1n}}{a_{11}} + a_{12} \frac{a_{1n}}{a_{12}} + \dots + a_{1n} \frac{a_{1n}}{a_{1n}} \\ a_{21} \frac{a_{1n}}{a_{11}} + a_{22} \frac{a_{1n}}{a_{12}} + \dots + a_{2n} \frac{a_{1n}}{a_{1n}} \\ \vdots \\ a_{n1} \frac{a_{1n}}{a_{11}} + a_{n2} \frac{a_{1n}}{a_{12}} + \dots + a_{nn} \frac{a_{1n}}{a_{1n}} \end{pmatrix}$$

Если рассмотреть i -ю строку получившегося вектора, и применив свойства совместности и обратной симметричности имеем:

$$a_{i1} \frac{a_{1n}}{a_{11}} + a_{i2} \frac{a_{1n}}{a_{12}} + \dots + a_{in} \frac{a_{1n}}{a_{1n}} = \frac{a_{11}}{a_{1i}} \frac{a_{1n}}{a_{11}} + \frac{a_{12}}{a_{1i}} \frac{a_{1n}}{a_{12}} + \dots + \frac{a_{1n}}{a_{1i}} \frac{a_{1n}}{a_{1n}} = \frac{a_{1n}}{a_{1i}} * n = w_i * n$$

Значит i элемент вектора получающегося в левой части уравнения $Aw = nw$ равен i элементу вектора стоящего справа. Это показывает справедливость (3) утверждения. Также заметим, что $w_i = a_{1n}/a_{1i} = a_{in}$, а значит искомый собственный вектор при условии выполнения условий (1)-(2) для матрицы A является её последним столбцом. ■

Данная теорема дает нам правило, по которому можно легко вычислять вектор приоритетов. Для этого требуется на основании некоторых элементов матрицы A построить её последний столбец, при условии выполнения условий (1)-(4).

В качестве элементов, однозначно определяющих матрицу парных сравнений не обязательно брать только элементы первой строки. Для того чтобы перечислить все такие наборы элементов, которые однозначно определяют матрицу парных сравнений, введем несколько определений.

Определение 2.2.1 Набор некоторых элементов матрицы $A = (a_{ij})_{n \times n}$ выше главной диагонали называется определяющим, если на его основе и с использованием свойств (2)-(4) можно однозначно определить все элементы матрицы A .

Определение 2.2.2 Пусть имеется некоторый набор элементов $a_{i_p j_p}$ матрицы A расположенных выше главной диагонали. Если существует среди этого набора такая тройка элементов $a_{i_a j_a}, b_{i_b j_b}, c_{i_c j_c}$, что $i_a = i_b, j_a = j_c, j_b = i_c$, то такой набор будем называть зависимым. В противном случае, будем говорить, что такой набор является независимым.

Определение 2.2.3 Минимальный по числу элементов независимый набор матрицы A будем называть базисным.

Определение 2.2.4 Рассмотрим неориентированный граф состоящий из n вершин. И пусть имеется какой-либо набор элементов $a_{i_p j_p}$ матрицы A . Если вершины i и j соединены ребром в том случае, когда в наборе $a_{i_p j_p}$ присутствует элемент $a_{i,j}$, то такой граф будем называть порожденным этим набором.

Теперь если рассмотреть неориентированный граф, построенный на некотором наборе n элементов матрицы A как на матрице смежности (присутствие в наборе элемента a_{ij} говорит о том, что вершины i и j соединены ребром), то можно убедиться в справедливости следующей теоремы:

Теорема 2.2.2 Для любого набора элементов $a_{i_p j_p}, p = 1, 2, \dots, n$ матрицы $A = (a_{ij})_{n \times n}$ расположенных выше главной диагонали, следующие высказывания эквивалентны:

1. Набор элементов $a_{i_p j_p}$ является базисным;
2. Набор $a_{i_p j_p}$ состоит из $n - 1$ элемента и порожденный им граф является связным;
3. Набор $a_{i_p j_p}$ состоит из $n - 1$ элемента и является независимым.

Данная теорема определяет правило выбора «определяющего» базисного набора, для построения которого нам потребуется всего $n - 1$ оценка эксперта. Это значительно меньше чем в обычном МАИ, в котором требуется $\frac{n(n-1)}{2}$ оценок, и в то же время позволяет нам построить матрицу парных сравнений, не лишенную свойства согласованности. К примеру, в качестве базисного набора можно взять элементы первой строки (схема сравнения с первым элементом), элементы выше или ниже какой-либо из диагоналей (последовательное сравнение). В принципе, в качестве базисного набора можно брать пары наиболее удобно сравниваемых элементов. Тогда эксперты будут выставять оценки выбирая для

себя те, которые проще всего определить, учитывая только выполнение свойства базисности получаемого набора.

При реализации метода анализа иерархий в случае нескольких экспертов, выставляющих оценки независимо друг от друга, можно слегка модифицировать исследуемый подход. Пусть каждый эксперт сравнивает элементы на основе своего базисного набора (выбирает наиболее удобный для него набор). Затем, рассчитав вектора приоритетов на основе данных, полученных от каждого эксперта, в качестве «окончательного» вектора можно взять их среднее значение. Это добавит решению объективности, при этом не утруждая экспертов сильнее чем обычно.

ГЛАВА 3

Теория нечетких множеств

В основе теории нечетких множеств лежит предположение о том, что произвольный элемент не обязательно может только содержаться или отсутствовать в каком либо множестве. Могут возникать ситуации когда вопрос принадлежности носит вероятностный характер.

Нечёткое множество — понятие, введённое Лотфи Заде в 1965 году в статье «Fuzzy Sets» в журнале Information and Control [3], в котором расширил классическое понятие множества, допустив, что функция принадлежности элемента множеству может принимать любые значения в интервале $[0,1]$, а не только значения 0 или 1.

Теории нечетких множеств и нечеткой логики позволяют расширить многие классы задач, придав им при этом более естественный характер. В дополнение к этому, во многих практических задачах могут возникнуть ситуации, когда нельзя найти детерминированное значение какого-либо параметра. В таком случае удобно применять механизмы «нечетких» множеств.

3.1 Основные понятия и определения

Определение 3.1.1 *Нечетким множеством A в некотором (непустом) пространстве X , что обозначается как $A \subseteq X$, называется множество пар*

$$A = \{(x, \mu_A(x)), x \in X\} \quad (3.1)$$

где

$$\mu_A : X \rightarrow [0, 1] \quad (3.2)$$

- функция принадлежности нечеткого множества A .

Функция (3.2) приписывает каждому элементу $x \in X$ степень его принадлежности к нечеткому множеству A , при этом можно выделить три случая:

1. $\mu_A(x) = 1$ означает полную принадлежность элемента x к нечеткому множеству A , то есть $x \in A$;
2. $\mu_A(x) = 0$ означает отсутствие принадлежности элемента x к нечеткому множеству A , то есть $x \notin A$;
3. $0 < \mu_A(x) < 1$ означает частичную принадлежность элемента x к нечеткому множеству A .

Заметим, что множество X может быть как дискретным, так и непрерывным (кусочно-непрерывным). И в том и в другом случае смысл функции принадлежности остается тем же.

3.2 Нечеткие числа LR типа

В качестве нечетких чисел в данной работе будем использовать числа LR типа, которые определяются следующим образом [6] :

Определение 3.2.1 *Функция L (или R) вида, это такая функция отображающая $R^+ \rightarrow [0, 1]$, для которой справедливо:*

1. $L(0) = 1$;
2. $L(x) < 1, \forall x > 0$;
3. $L(x) > 0, \forall x < 1$;
4. $L(1) = 1$;
5. L убывает на промежутке $(0, 1)$;

Определение 3.2.2 *Нечеткое число ξ является числом LR типа, если существуют такие функции L и R вида и некоторые числа α, β, t , такие, что;*

$$\mu_\xi(x) = \begin{cases} L(\frac{m-x}{\alpha}) & , x \leq m \\ R(\frac{x-m}{\beta}) & , x \geq m \end{cases} \quad (3.3)$$

где m - называется модой числа, а α и β соответственно левой и правой границей нечеткости.

Определение 3.2.3 Нечеткое число ξ называется интервалом LR типа, если существуют такие функции L и R вида и некоторые числа α, β, a, b , такие, что;

$$\mu_{\xi}(x) = \begin{cases} L(\frac{m-x}{\alpha}) & , x < a \\ 1 & , a \leq x \leq b \\ R(\frac{x-m}{\beta}) & , x > b \end{cases} \quad (3.4)$$

где отрезок $[a, b]$ - называется ядром интервала, а α и β соответственно левой и правой границей нечеткости.

Таким образом, мы можем записать нечеткое число LR типа при заданных L и R функциях как упорядоченную тройку чисел:

$$\mu_{\xi}(x) = (m, \alpha, \beta)_{L,R} \quad (3.5)$$

а нечеткий интервал LR типа как упорядоченную четверку чисел:

$$\mu_{\xi}(x) = (a, b, \alpha, \beta)_{L,R} \quad (3.6)$$

Заметим, что нечеткие числа LR типа дают вполне естественную формализацию «неуверенной» классификации какого-либо числа. Числа, которые мы привыкли оценивать как «число, с некоторой долей уверенности находящееся между a и b » или «число, скорее всего равное m » можно представить в виде $(\frac{a+b}{2}, a, b)_{L,R}$ и $(m, \alpha, \beta)_{L,R}$ соответственно.

3.3 Примеры LR чисел

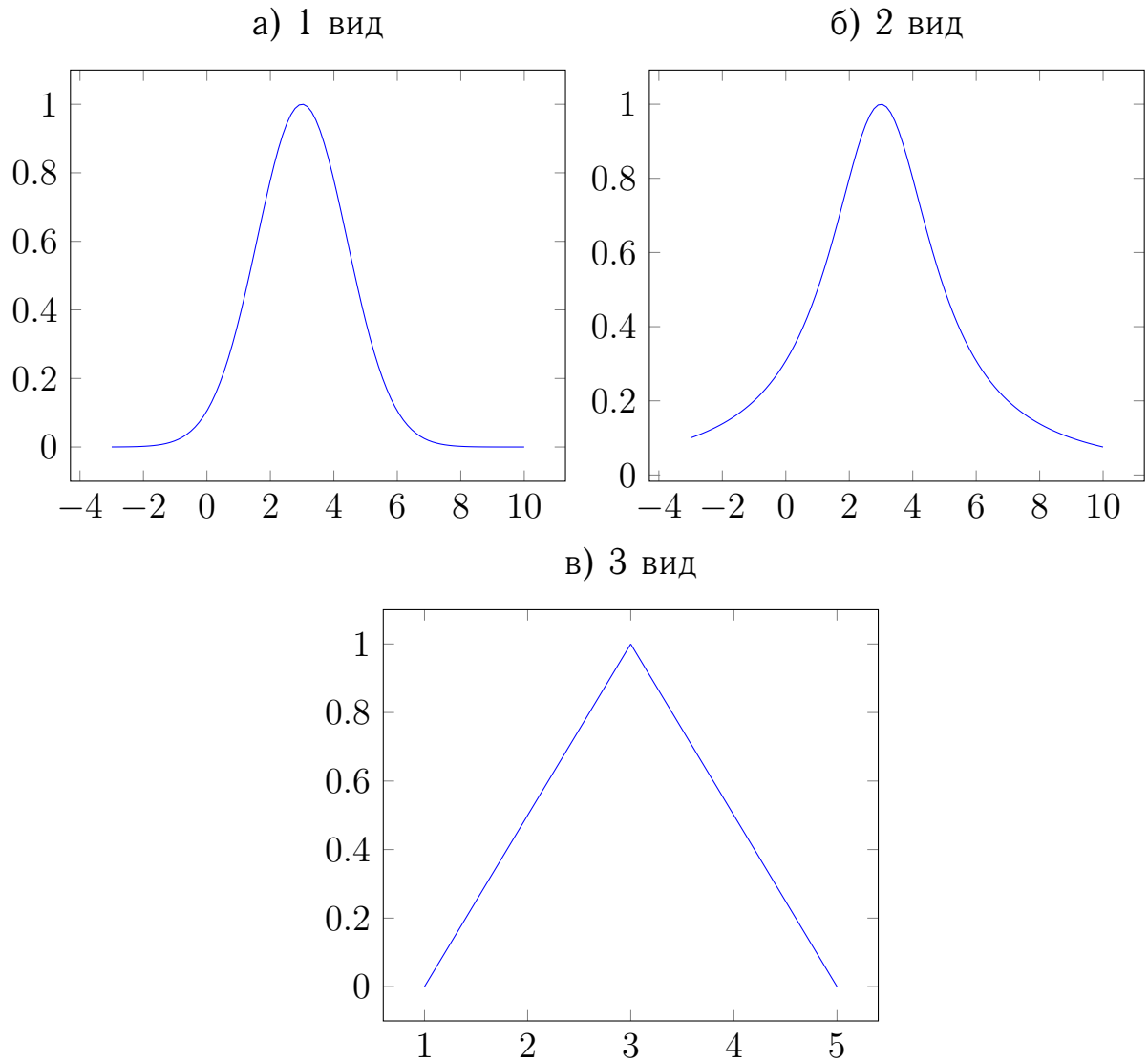
Внешний вид и свойства числа LR типа определяются его L и R функциями и их параметрами. В основном, в качестве $L(R)$ функций используют следующие выражения:

1. $f(x) = e^{-|x|^p}$;
2. $f(x) = \frac{1}{1+|x|^p}$;
3. $f(x) = 1 - x$;

где p - произвольный вещественный параметр.

При использовании таких $L(R)$ функций, графики функций принадлежности выглядят следующим образом:

Рис. 3.1: Графики функций описанных в 3.3 с параметрами $p = 2, m = 3, \alpha = 2, \beta = 2$.



3.4 Операции с нечеткими числами и интервалами

Выведем теперь правила выполнения арифметических операций с нечеткими числами [8]. Рассмотрим правые (R) функции каких либо двух нечетких чисел LR типа:

$$\mu_{\xi_1}(x) = R\left(\frac{x_1 - m_1}{\beta_1}\right); \mu_{\xi_2}(x) = R\left(\frac{x_2 - m_2}{\beta_2}\right)$$

Заметим, что эти формулы можно переписать в следующем виде:

$$x_1^* = m_1 + \beta_1 R^{-1}(\mu^*); x_2^* = m_2 + \beta_2 R^{-1}(\mu^*)$$

Тогда для некоторого z^* справедливо:

$$z^* = x_1^* + x_2^* = (m_1 + m_2) + (\beta_1 + \beta_2) R^{-1}(\mu^*)$$

откуда

$$\mu_r(z^*) = \mu^* = R\left(\frac{z^* - m_3}{\beta_3}\right)$$

где $m_3 = m_1 + m_2, \beta_3 = \beta_1 + \beta_2$. Проведя аналогичные рассуждения для L функций, получаем что:

$$z^* = x_1^* + x_2^* = (m_1 + m_2, \alpha_1 + \alpha_2, \beta_1 + \beta_2)_{L,R} \quad (3.7)$$

Аналогичным образом получаем,

для вычитания:

$$z^* = x_1^* - x_2^* = (m_1 - m_2, \alpha_1 + \beta_2, \beta_1 + \alpha_2)_{L,R} \quad (3.8)$$

для умножения:

$$z^* = x_1^* * x_2^* = (m_1 m_2, m_1 \alpha_2 + m_2 \alpha_1, m_2 \beta_1 + m_1 \beta_2)_{L,R} \quad (3.9)$$

для получения числа, обратного данному:

$$z^* = z^{*-1} = (1/m, \beta/m^2, \alpha/m^2)_{L,R} \quad (3.10)$$

для деления двух чисел:

$$z^* = x_1^* / x_2^* = \left(\frac{m_1}{m_2}, \frac{m_1 \beta_2 + m_2 \alpha_1}{m_2^2}, \frac{m_1 \alpha_1 + m_2 \beta_1}{m_2^2}\right)_{L,R} \quad (3.11)$$

Проведя схожие рассуждения для нечетких интервалов, можно вывести следующие формулы арифметических операций:

Сложение:

$$z^* = x_1^* + x_2^* = (a_1 + a_2, b_1 - b_2, \alpha_1 + \alpha_2, \beta_1 + \beta_2)_{L,R} \quad (3.12)$$

Вычитание:

$$z^* = x_1^* - x_2^* = (a_1 - a_2, b_1 + b_2, \alpha_1 + \beta_2, \beta_1 + \alpha_2)_{L,R} \quad (3.13)$$

Умножение:

$$z^* = x_1^* * x_2^* = (a_1 a_2, b_1 b_2, a_1 \alpha_2 + a_2 \alpha_1, b_1 \beta_2 + b_2 \beta_1)_{L,R} \quad (3.14)$$

Деление:

$$z^* = x_1^* / x_2^* = \left(\frac{a_1}{b_2}, \frac{b_1}{a_2}, \frac{a_1 \beta_2 + b_2 \alpha_1}{b_2^2}, \frac{b_1 \alpha_2 + a_2 \beta_1}{a_2^2} \right)_{L,R} \quad (3.15)$$

3.5 Дефаззификация

Дефаззификацией называется процесс приведения нечеткого числа к его вещественной интерпретации. В теории нечётких множеств дефаззификация аналогична нахождению характеристик положения случайных величин (математического ожидания, моды, медианы) в теории вероятности. Простейшим способом дефаззификации является выбор чёткого числа с максимальной степенью принадлежности. Более сложные методы основываются на идее поиска точки сгущения (центра масс) области, расположенной между графиком функции принадлежности и осью абсцисс. Это делается с целью «усреднить» возможные принимаемые нечетким числом значения, с учетом соответствующих им степеней принадлежности.

Рассмотрим два наиболее часто используемых метода:

1. Метод центра масс:

$$a_r = \frac{\int_{min}^{max} u \mu_\xi(u) du}{\int_{min}^{max} \mu_\xi(u) du}, \quad (3.16)$$

2. Метод медианы:

$$a_r : \int_{min}^{a_r} \mu_\xi(u) du = \int_{a_r}^{max} \mu_\xi(u) du, \quad (3.17)$$

Где min, max - минимальные и максимальные «значимые» границы области значений для функции принадлежности. Выбираются они произвольно, в зависимости от требуемой точности вычислений и вида графика функции принадлежности нечеткого числа.

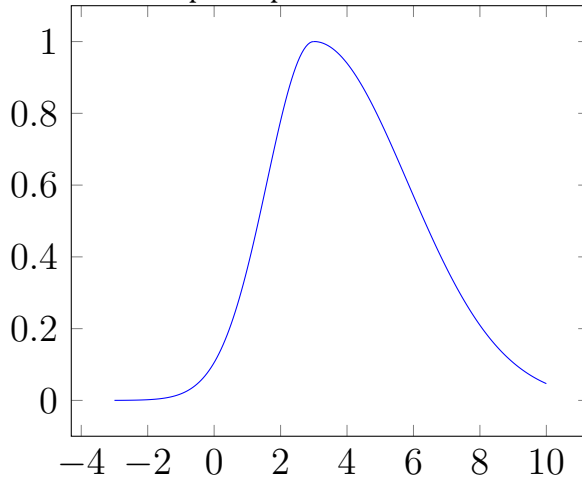
Пример:

Пусть мы имеем нечеткое число LR типа, с функцией принадлежности вида:

$$\mu_{\xi}(x) = \begin{cases} e^{-|(\frac{3-x}{2})|^2} & , x < 3 \\ e^{-|(\frac{x-3}{4})|^2} & , x \geq 3 \end{cases}$$

График этой функции представлен на рисунке 3.2 Такое нечеткое мно-

Рис. 3.2: Пример нечеткого числа.



жество можно описать как «почти 3». Рассмотрим результаты его дефаззификации предложенными ранее методами:

1. Метод центра масс: $a_r = 3.9458$,
2. Метод медианы: $a_r = 3.8764$,

Как видно из примера, за счет R функции, график $\mu_{\xi}(x)$ после точки $x = 3$ убывает медленнее, чем до неё. За счет этого, большая часть площади фигуры заключенной между $\mu_{\xi}(x)$ и осью абсцисс, расположена правее точки $x = 3$. Это привело к тому, что значение после дефаззификации также сместилось в правую сторону от $x = 3$.

ГЛАВА 4

Упрощенный вариант метода анализа иерархий в условиях нечетких данных

В идее этого метода лежит применение теории нечетких множеств к построению экспертных оценок. Так как оценки экспертов по своей сути носят субъективный характер, причем зачастую не всегда точно определенный (эксперт может сомневаться в выставленной им оценке). То, чтобы снизить влияние человеческой ошибки, и позволить эксперту оперировать не только точными оценками, но и «интервалами» оценивания, оправдано будет использовать результаты теории нечетких множеств. Опираясь на «нечеткие» данные, мы построим нечеткий вектор весов $\Psi = (\psi_1, \psi_2, \dots, \psi_n)$, произведем его дефаззификацию и нормировку.

4.1 Экспертные оценки

Согласно упрощенному МАИ, для решения поставленной задачи нам достаточно всего $(n - 1)$ экспертной оценок, представляющих собой базисный набор элементов матрицы A . Выбрав базисный набор (все элементы из любой строки или столбца, элементы выше главной диагонали и т.д.) появляется задача оптимального в рамках текущей задачи выбора метода оценивания. С учетом инструментов нечеткой логики и классического МАИ мы имеем следующие возможности:

1. *Фаззификация имеющихся оценок.* В таком случае оценки можно выставлять в соответствии с классическим МАИ, но в таком случае требуется только добавить «уровни доверия» этой оценке. Эти уровни в дальнейшем следует использовать как левую и правую границу нечеткости для оценок. В качестве медианы использовать само значение оценки выставленное экспертом. Такой подход позволит снизить влияние субъективности, оставляя экспертов в условиях классического МАИ. Шкалу Саати в таком случае предлагается понимать не как дискретную, а как непрерывную. К тому же

Таблица 4.1: Примеры оценок экспертов

Оценка эксперта	Нечеткое значение
Практически равны	$(1/2, 1, 2)$
Приблизительно в n раз более предпочтительно	$(n - 1, n, n + 1)$
Приблизительно в n раз менее предпочтительно	$(1/(n + 1), 1/n, 1/(n - 1))$
Между k и n раз более предпочтительно	$(k, (k + n)/2, n)$
Между k и n раз менее предпочтительно	$(1/n, 2/(k + n), 1/k)$

допустимо использовать любую количественную шкалу. Или можно использовать некоторую «интервальную» оценку, когда эксперт выбирает между какими значениями скорее всего лежит точная оценка. Примеры возможных оценок приведены в таблице 4.1.

2. *Нечеткая дискретизация оценки.* В этом случае предлагаем использовать в качестве отправного вектора оценок дискретную шкалу Саати. От эксперта же требуется выставить для каждой оценки из шкалы её значение функции принадлежности - вещественное число от 0 до 1. К примеру, это может выглядеть следующим образом:

$$a_{ij} = \begin{matrix} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{matrix} & 0 & 0 & 0.05 & 0.2 & 0.8 & 0.2 & 0.05 & 0 & 0 \end{matrix}$$

В таком случае, мы сможем учесть возможность сомнения эксперта в выставленной оценке.

3. *Нечеткий интервал оценки.* Такой подход имеет в основе размышление вида: «эта оценка лежит в интервале от a до b с уровнями доверия α и β ». От эксперта в таком случае требуется определить, в каком диапазоне лежит по его ощущениям текущая оценка и какова его уверенность в этом. Такой подход позволит снизить субъективность, и дать больше простора выбора эксперту. Границы диапазона предлагается использовать как в шкале Саати. И как и в первом случае, можно вместо шкалы Саати использовать любую количественную шкалу.

В первом и в третьем случае мы получаем нечеткое число и интервал LR типа соответственно. Во втором случае мы получаем классическое дискретное нечеткое множество. Так как, по сути, первый и третий варианты являются непрерывным аналогом второго, позволяющим упростить процесс выполнения арифметических операций, то в данной работе мы будем рассматривать только первый и третий вариант.

4.2 Построение вектора весов

Используя результаты второй и третьей главы, опишем алгоритм построения искомого вектора весов. Требуется на основе какого либо набора элементов матрицы оценок A определить, является ли он базисным, и в случае базисности, посчитать на его основе нечеткий вектор весов. Данный алгоритм лежит в основе программной реализации, поэтому опишем его подробнее.

Пусть мы имеем некоторый набор экспертных оценок $a_{i_p j_p}$. Проверим этот набор на базисность. Это можно сделать простым перебором всех возможных троек элементов, и проверкой их на независимость согласно определению 2.2.2. С другой стороны, это можно проверить за один проход по всем элементам массива. Чтобы показать это, введем понятие «запрещенных» индексов - индексов, которые при появлении среди i_p или j_p в нашем наборе делают его связным. Теперь, пусть мы идем перебором по элементам набора. Если какой-то индекс строки (столбца) встречается более чем один раз, то этот индекс становится «запрещенным» для появления среди номеров столбцов (строк) в наборе. Ведь в противном случае появляется зависящая тройка элементов. Таким образом, перебирая элементы и запоминая «запрещенные» мы сможем проверить базисность за $O(n)$ действий.

Теперь, имея базисный набор, нам необходимо построить вектор «нечетких» весов. Для удобства дальнейших рассуждений, сформулируем теорему:

Теорема 4.2.1 *Любую компоненту вектора приоритетов можно по-*

лучить как произведение не более чем $n - 1$ элемента базисного набора.

Доказательство. Сначала вспомним, что подразумевается под графом порожденным матрицей смежности. Пусть мы имеем матрицу $n \times n$ и граф из n вершин, пронумерованных от 1 до n . Вершины i и j соединены ребром, если элемент матрицы a_{ij} или a_{ji} отличен от нуля. Причем такой граф будем считать неориентированным. Согласно теореме 2.2.2, граф, порожденный нашим набором должен быть связным и без циклов. Тогда из вершины с номером n выходит как минимум 1 ребро (иначе нарушится связность). Также, благодаря связности графа мы из вершины n можем попасть в любую из оставшихся не более чем за $n - 1$ шагов.

Пусть из вершины n выходит только одно ребро и соединяет оно с вершиной i . Та, в свою очередь, соединена с вершиной j . Значит в наборе присутствует 2 элемента: a_{in} и a_{ji} (порядок индексов не важен, так как благодаря свойству симметричности матрицы A : $a_{in} = 1/a_{ni}$). На основе свойства совместности имеем: $a_{jn} = a_{ji} * a_{in}$. Так как из вершины j или i выходит еще минимум 1 ребро, аналогичным образом получаем некоторый элемент a_{kn} . Проходя таким образом по графу в глубину (максимальная глубина этого графа $= n - 1$), мы последовательно получаем все элементы вида a_{*n} в виде произведения элементов соответствующих ребрам, по которым мы проходим. Эти элементы являются компонентами последнего вектора матрицы A , и согласно теореме 2.2.1 этот вектор и будет искомым вектором весов.

Если из вершины с номером n выходит более одного ребра, то рассмотрев аналогично предыдущему случаю каждую из ветвей выходящих из неё, также получим все элементы последнего вектора. ■

К примеру, пусть мы имеем следующий базисный набор:

$$a_{ij} = \begin{pmatrix} 1 & + & * & * & * & * \\ * & 1 & * & + & * & * \\ * & * & 1 & * & + & * \\ * & * & * & 1 & * & + \\ * & * & * & * & 1 & + \\ * & * & * & * & * & 1 \end{pmatrix}$$

Здесь * - элементы, не входящие в набор, + - входящие, а 1 - элементы на главной диагонали. Тогда граф, порожденный этим набором будет выглядеть как показано на Рис. 4.1.

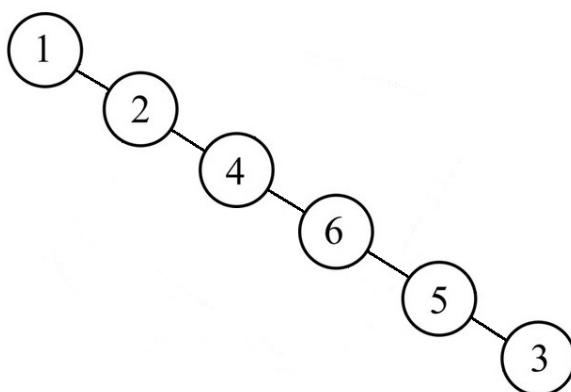


Рис. 4.1: Граф, порожденный базисным набором.

4.3 Дефаззификация

Найденный вектор весов представляет собой вектор нечетких чисел выбранного нами вида. Для завершения вычислений необходимо его дефаззифицировать. Сделать это можно любым из предложенных ранее способов. Для программной реализации реализованы несколько численных методов интегрирования (метод прямоугольников, метод трапеций и метод Симпсона). Но применимы в данном случае любые более удобные или более предпочтительные методы. После дефаззификации и нормировки полученных значений мы получаем искомый вектор весов.

ГЛАВА 5

Анализ результатов

Предложенный в данной работе подход будем сравнивать с различными уже существующими вариациями МАИ. В качестве исходных данных будем использовать пример, описанный в [14]. Методы используемые для сравнения:

1. Классический МАИ
2. Упрощенный МАИ
3. Классический МАИ с применением теории нечетких множеств
4. Предложенный нами метод.

5.1 Подготовка данных

Чтобы провести анализ и сравнение методов, надо их поместить в равные условия. Для этого мы возьмем данные из статей [14] и [9]. В этих работах рассмотрен один и тот же прикладной пример метода анализа иерархий. С единственным отличием в том, что в [9] используются нечеткие экспертные оценки, а в [14] классический МАИ. Для использования упрощенных вариаций МАИ, в первоначальном сравнении мы будем использовать схему «сравнения с первым». Далее сравним результаты, получаемые при использовании различных базисных наборов и при их комбинировании.

5.2 Проблема водного хозяйства в Китае

Для реализации примера, будет использоваться задача оптимального распределению инвестиций правительства в развитие водных бассейнов Китая. В качестве одной из инфраструктурных отраслей народного хозяйства, водное хозяйство играет большую роль для его устойчивого развития. Основные проекты развития водного хозяйства включают:

- Проекты водных резервуаров
- Ирригационные проекты
- Проект предотвращения и борьбы с наводнениями
- Проект контроля заболачиваемости
- Проект водоснабжения
- Гидроэнергетические проекты
- Проект сохранения воды и почв

Таким образом, мы можем видеть, что функции и цели охраны природы связаны с водными ресурсами, и это является важной составляющей для гарантии сохранения экономического развития и улучшения качества жизни людей.

В социально-экономическом развитии Китая в течение последних шестидесяти лет защита водного хозяйства играет большую роль для его охраны и защиты ресурсов, социальной стабильности и экосистемы. Проекты охраны водных ресурсов направлены на благо общества, и зачастую являются долгосрочными. То есть, большие объемы инвестиций производятся в одно время, а окупаемость капиталовложений проходит, как правило, долго. Такие проекты, обычно, мало интересуют частных инвесторов. Следовательно, большая часть инвестиций в развитие водного хозяйства и капитального строительства происходит от правительства.

Существует много других областей, заслуживающих не меньшего количества инвестиций от правительства (например такие как образование). А количество выделяемых на такие проекты денег ограничено, чтобы не затормозить развитие национальной экономики. Поэтому необходимо найти наиболее эффективный способ выделить проекты развития водного хозяйства среди различных областей.

Таким образом, главным целевым критерием является разумное распределение инвестиций. Критериями первого уровня будут описанные ранее проекты развития водного хозяйства. Ну а сравниваемыми

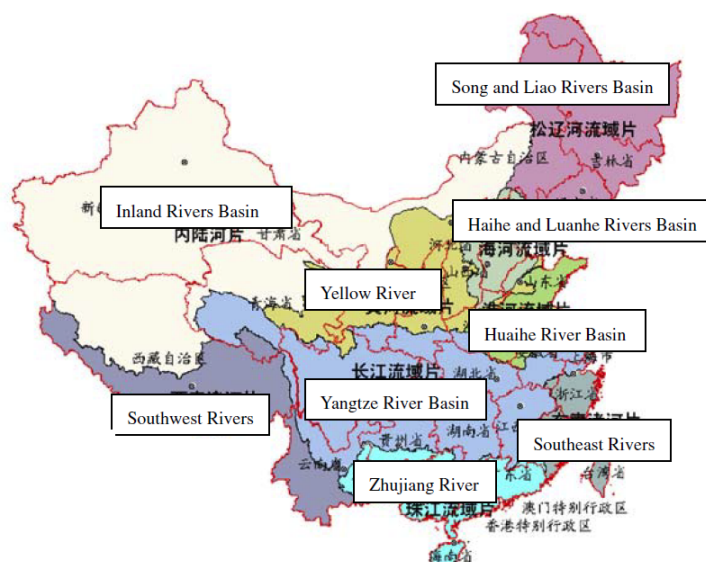


Рис. 5.1: Водные бассейны Китая.

альтернативами будут самые крупные водные бассейны Китая (см. Рис. 5.1). Визуально данная система иерархий представлена на рис. 5.2

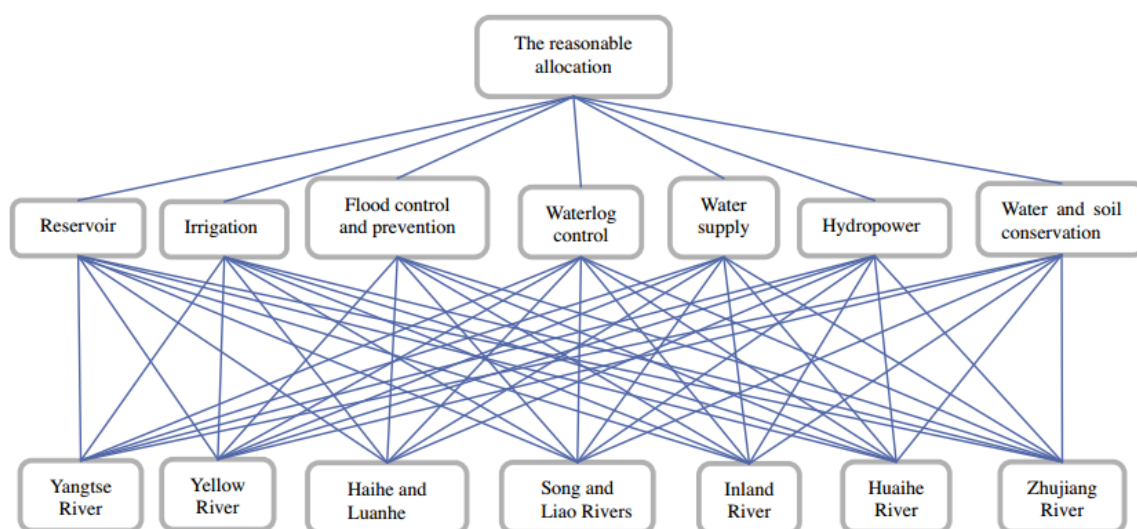


Рис. 5.2: Иерархия распределения инвестиций.

5.3 Результаты работы различных методов

Рассмотрим теперь результаты, полученные при реализации четырех различных модификаций метода анализа иерархий. Далее будут использоваться сокращения: МАИ - метод анализа иерархий, УМАИ -

упрощенный метод анализа иерархий, НМАИ - метод анализа иерархий с применением теории нечетких множеств, УНМАИ - упрощенный метод анализа иерархий с применением теории нечетких множеств.

Таблица 5.1: Вектор приоритетов для проектов охраны водных ресурсов

Приоритеты проектов	МАИ	УМАИ	НМАИ	НУМАИ
Проекты водных резервуаров	0,1302	0,1130	0,1217	0,1167
Ирригационные проекты	0,0524	0,0452	0,0852	0,0472
Проект предотвращения и борьбы с наводнениями	0,3798	0,3767	0,3895	0,3851
Проект контроля заболачиваемости	0,0524	0,0576	0,0568	0,0372
Проект водоснабжения	0,2668	0,2925	0,2191	0,2877
Гидроэнергетические проекты	0,0343	0,0382	0,0426	0,0340
Проект сохранения воды и почв	0,0843	0,0765	0,0852	0,0918

Таблица 5.2: Вектор приоритетов водных бассейнов по критерию проектов водных резервуаров

Проекты водных резервуаров	МАИ	УМАИ	НМАИ	НУМАИ
Янцзы	0,4012	0,4631	0,4487	0,3754
Хуанхэ	0,0585	0,0771	0,0851	0,0951
Хайхэ и Луаньхэ	0,0264	0,0578	0,0638	0,0509
Ляохэ и Сон	0,2216	0,1323	0,1702	0,2228
Внутренние воды	0,0973	0,0926	0,0774	0,1023
Хуайхэ	0,0973	0,0842	0,0774	0,0935
Чжуцзян	0,0973	0,0926	0,0774	0,0597

Таблица 5.3: Вектор приоритетов водных бассейнов по критерию ирригационных проектов

Ирригационные проекты	МАИ	УМАИ	НМАИ	НУМАИ
Янцзы	0,0648	0,0601	0,0938	0,0471
Хуанхэ	0,3105	0,2706	0,3343	0,3067
Хайхэ и Луаньхэ	0,1416	0,1806	0,1290	0,1818
Ляохэ и Сон	0,0942	0,0802	0,0909	0,1250
Внутренние воды	0,2149	0,2106	0,1935	0,1967
Хуайхэ	0,1416	0,1806	0,1232	0,1318
Чжуцзян	0,0321	0,0171	0,0352	0,0105

Таблица 5.4: Вектор приоритетов водных бассейнов по критерию проекта предотвращения и борьбы с наводнениями

Проект предотвращения и борьбы с наводнениями	МАИ	УМАИ	НМАИ	НУМАИ
Янцзы	0,3678	0,3967	0,3399	0,3829
Хуанхэ	0,1023	0,0793	0,1034	0,0883
Хайхэ и Луаньхэ	0,0375	0,0566	0,0443	0,0448
Ляохэ и Сон	0,1574	0,1586	0,2069	0,1978
Внутренние воды	0,0202	0,0440	0,0296	0,0292
Хуайхэ	0,1574	0,1322	0,1379	0,1365
Чжуцзян	0,1574	0,1322	0,1379	0,1202

Таблица 5.5: Вектор приоритетов водных бассейнов по критерию проекта контроля заболачиваемости

Проект контроля заболачиваемости	МАИ	УМАИ	НМАИ	НУМАИ
Янцзы	0,3337	0,3327	0,2883	0,3266
Хуанхэ	0,0421	0,0532	0,0499	0,0478
Хайхэ и Луаньхэ	0,0421	0,0532	0,0499	0,0478
Ляохэ и Сон	0,1624	0,1365	0,1163	0,1773
Внутренние воды	0,0212	0,0414	0,0253	0,0316
Хуайхэ	0,2358	0,2485	0,3540	0,2511
Чжуцзян	0,1624	0,1242	0,1163	0,1176

Таблица 5.6: Вектор приоритетов водных бассейнов по критерию проекта водоснабжения

Проект водоснабжения	МАИ	УМАИ	НМАИ	НУМАИ
Янцзы	0,0436	0,0416	0,0815	0,0299
Хуанхэ	0,2080	0,1667	0,2060	0,1864
Хайхэ и Луаньхэ	0,3051	0,3096	0,3090	0,2922
Ляохэ и Сон	0,1331	0,1467	0,1202	0,1464
Внутренние воды	0,1331	0,1467	0,1202	0,1464
Хуайхэ	0,1331	0,1467	0,1202	0,1464
Чжуцзян	0,0436	0,0416	0,0429	0,0319

Таблица 5.7: Вектор приоритетов водных бассейнов по критерию гидроэнергетических проектов

Гидроэнергетические проекты	МАИ	УМАИ	НМАИ	НУМАИ
Янцзы	0,4517	0,4442	0,4590	0,4351
Хуанхэ	0,0798	0,0807	0,0920	0,0945
Хайхэ и Луаньхэ	0,0300	0,0538	0,0462	0,0529
Ляохэ и Сон	0,1149	0,0968	0,1104	0,1227
Внутренние воды	0,0300	0,0538	0,0407	0,0529
Хуайхэ	0,0564	0,0691	0,0678	0,0758
Чжуцзян	0,2367	0,2014	0,1840	0,1657

Таблица 5.8: Вектор приоритетов водных бассейнов по критерию проекта сохранения воды и почв

Проект сохранения воды и почв	МАИ	УМАИ	НМАИ	НУМАИ
Янцзы	0,2409	0,2571	0,2466	0,2275
Хуанхэ	0,3377	0,3673	0,3699	0,3539
Хайхэ и Луаньхэ	0,1693	0,1285	0,1438	0,1444
Ляохэ и Сон	0,0481	0,0571	0,0514	0,0634
Внутренние воды	0,0777	0,0734	0,0685	0,0766
Хуайхэ	0,0777	0,0734	0,0685	0,0766
Чжуцзян	0,0481	0,0435	0,0428	0,0473

5.4 Сравнение и анализ методов

Рассмотрим теперь окончательный вектор приоритетов для сравниваемых водных бассейнов по критерию количества требуемых инвестиций. Как получить этот вектор посредством линейной свертки описано в первой главе данной работы. На основе данных из таблиц 5.1 - 5.8 получаем конечные вектора приоритетов по целевому критерию (Таблица 5.9).

Как видно из Таблицы 5.9, окончательный вектор приоритетов в каждой из реализаций выглядит примерно одинаково. Для большей наглядности визуализируем эти вектора в качестве круговой диаграммы (Рис. 5.3).

В данном примере, в качестве независимого набора были выбраны элементы первой строки. А нечеткие оценки экспертов выставлялись

Таблица 5.9: Вектор приоритетов водных бассейнов по целевому критерию

Значимость для инвестиций	МАИ	УМАИ	НМАИ	НУМАИ
Янцзы	0,2474	0,2520	0,2698	0,2303
Хуанхэ	0,1550	0,1255	0,1625	0,1365
Хайхэ и Луаньхэ	0,1275	0,1404	0,1208	0,1337
Ляохэ и Сон	0,1474	0,1301	0,1511	0,1596
Внутренние воды	0,0775	0,0886	0,0728	0,0854
Хуайхэ	0,1389	0,1400	0,1288	0,1406
Чжуцзян	0,1060	0,0735	0,0944	0,1063

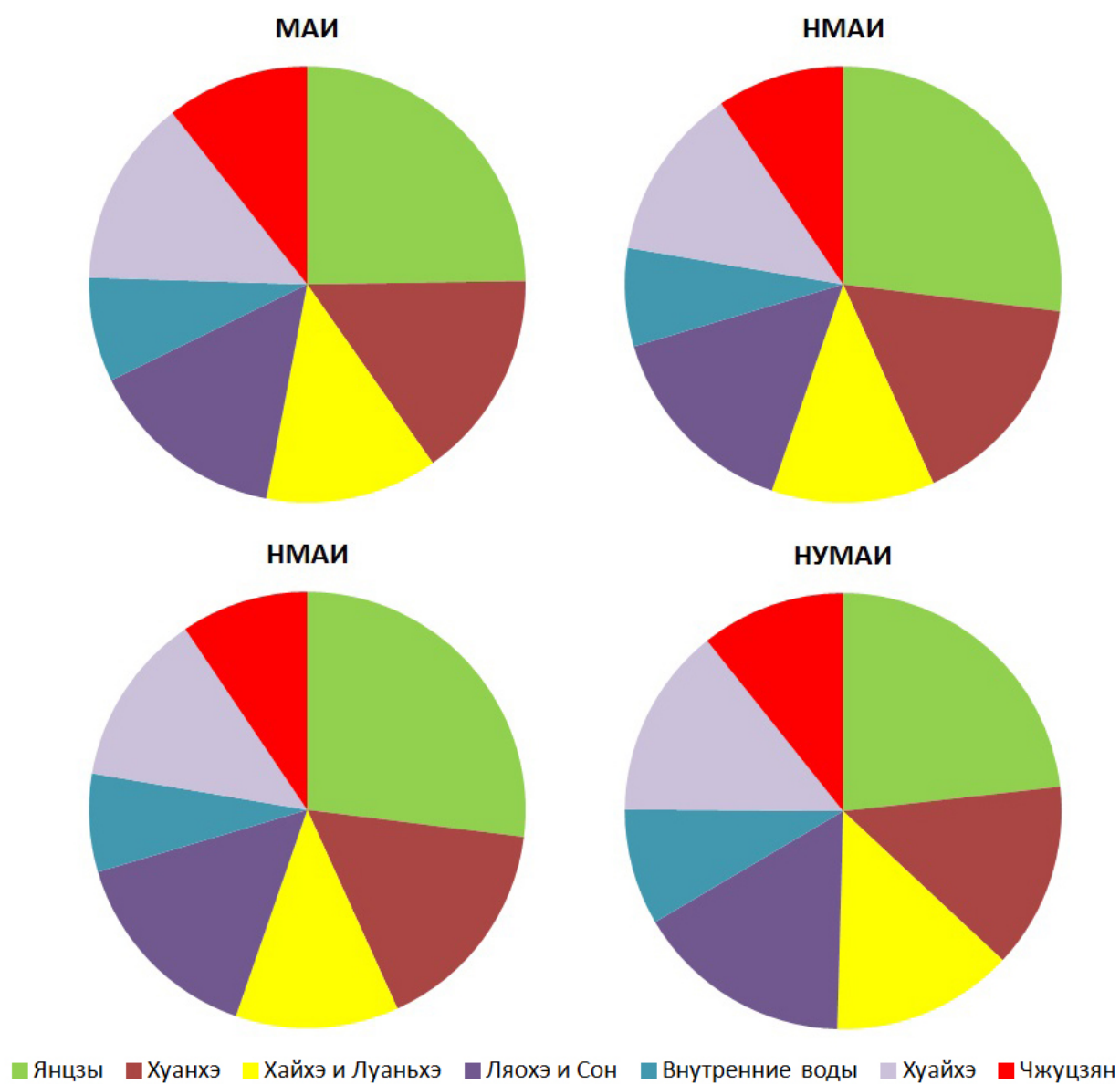


Рис. 5.3: Круговая диаграмма векторов приоритетов.

следующим образом: оценка из классического МАИ бралась в качестве медианы m нечеткого числа, а левая и правая граница брались равными $m - 1$ и $m + 1$ соответственно. В качестве функций LR типа использовалась функция: $f(x) = \frac{1}{1+|x|^p}$. Дефаззификация производилась по правилу центра масс.

Примем классический МАИ в качестве «эталонного» метода. Рассмотрим максимальное и среднее отклонение сравниваемых подходов. Результаты этого сравнения представлены в Таблице 5.10.

Таблица 5.10: Отклонения модификаций МАИ от «эталонного» метода. МО и МО% - это максимальные отклонения (в обычном виде и процентном соотношении). СО и СО% - средние отклонения.

Метод	МО	МО%	СО	СО%
УМАИ	0,0325	30,66	0,0155	12,65
НМАИ	0,0224	10,94	0,0095	7,11
НУМАИ	0,0185	11,94	0,0091	6,24

Как видно из Таблицы 5.10, наиболее близкими к МАИ оказались методы НМАИ и НУМАИ. В тоже время, УМАИ, оказавшись методом, наиболее отличным от МАИ в результатах, тоже дает вполне применимые на практике данные. На круговых диаграммах видно, что несмотря на различия в векторах приоритетов, в целом распределение весов вышло очень похожим.

Справедливо будет заметить, что брать в качестве «эталона» классический МАИ не совсем корректно, так в большинстве случаев имеется расхождение результатов, полученных с помощью МАИ, и реальных данных. Методы анализа иерархий применяются в задачах, когда хочется избежать расчетов реальных зависимостей альтернатив, или же когда «предпочтительность» одного элемента над другим точно не измерима. Из-за этого, данные получаемые с помощью МАИ носят приблизительный и субъективный характер. А при наличии большой «модельной» ошибки МАИ и вовсе становится неприменимым.

За счет этих замечаний становится ясно, что главная задача какой-либо модификации метода анализа иерархий состоит не в том, чтобы максимально приблизиться по результатам к классической его вариации.

Основным критерием является приемлемое отображение действительности.

Основная проблема МАИ заключается в невыполнении свойства согласованности для матрицы парных сравнений. За счет этого, если максимальное собственное число сильно превышает значение размерности матрицы сравнений, то полученные в итоге результаты нельзя назвать приемлемыми. С другой стороны, УМАИ и НУМАИ основываются на полностью согласованной матрице парных сравнений, что избавляет их от этой проблемы. И корректность результатов в таком случае полностью зависит от адекватности выставляемых экспертами оценок.

Также стоит отметить, что упрощенные вариации МАИ позволяют проводить исследования в условиях, отличных от классического метода. За счет меньшего числа требуемых оценок, мы можем реализовать анализ иерархий с использованием различных базисных наборов, а затем усреднить полученные при этом результаты. Эксперты при этом должны будут выставить свои оценки в соответствии с каждым применяемым набором. Или же, упрощенный анализ иерархий можно реализовать таким образом, чтобы эксперты сами выбирали себе удобный для них набор базисных элементов.

Помимо не меньшей адекватности получаемых результатов, каждая из модификаций позволяет избавиться от некоторых недостатков МАИ:

1. УМАИ - Существенно снижает количество производимых вычислений. В то же время требует уделения большего внимания к выставленным экспертным оценкам.
2. НМАИ - Позволяет ставить экспертам их оценки более свободно, избавляя от необходимости выбора какого-либо конкретного значения. В тоже время увеличивается количество производимых вычислений.
3. НУМАИ - Является объединением предыдущих двух модификаций, позволяющая при меньшем количестве «нечетких» оценок получать результаты, не хуже чем в классическом МАИ.

ЗАКЛЮЧЕНИЕ

В данной работе были рассмотрены некоторые существующие модификации метода анализа иерархий, а именно:

1. Классический метод анализа иерархий.
2. Упрощенный метод анализа иерархий.
3. Метод анализа иерархий с применением теории нечетких множеств.

Также предложен новый подход, являющийся объединением 2 и 3 методов. Для упрощения реализации данного метода было доказано несколько утверждений.

В ходе работы показано, что новая методика ничем не уступает уже существующим, а в некоторых аспектах и превосходящая их. В частности, все четыре метода были реализованы для реально существующего, актуального примера, и произведен их сравнительный анализ. Проведенный анализ наглядно показал применимость всех четырех методов, и аналогичность результатов, получаемых при их реализации на одинаковых данных.

Для изучения предлагаемого нового подхода сделана его компьютерная реализация на языке C#. Исходных код основных компонентов написанной программы можно найти в приложении к данной работе.

СПИСОК ЛИТЕРАТУРЫ

1. Саати Т. Л. Принятие решений. Метод анализа иерархий. — М.: Радио и связь, 1989. — 316 с.
2. Ногин В.Д. Упрощенный вариант метода анализа иерархий на основе нелинейной свертки критериев// ЖВМиМФ, 2004, т. 44, № 7, С. 1259-1268.
3. Заде Л. Понятие лингвистической переменной и его применение к принятию приближенных решений. — М.: Мир, 1976. — 166 с.
4. Кофман А. Введение в теорию нечетких множеств. — М.: Радио и связь, 1982. — 432 с.
5. Saaty T.L. An eigenvalue allocation model for prioritization and planning// Energy Management and Policy Center, University of Pennsylvania, 1972
6. D. Dubois and P. Prade. Operations on fuzzy numbers. International Journal of Systems Science, 9(6):613–626, 1978. – 56 с.
7. D. Dubois and P. Prade. The mean value of a fuzzy number. Fuzzy Sets and Systems, 320 24(3):279–300, 1987.
8. A. Alim, F. T. Johora, S. Babu, A. Sultana Elementary Operations on L-R Fuzzy Number. Advances in Pure Mathematics, Vol.05 No.03(2015)
9. Bin Zhu, Zeshui Xu. Analytic hierarchy process-hesitant group decision making. European Journal of Operational Research 239 (2014) 794–801
10. M. B. Javanbarg, C. Scawthorn, J. Kiyono, B. Shahbodaghkhan Fuzzy AHP-based multicriteria decision making systems using particle swarm optimization. Expert Systems with Applications 39 (2012) 960–966

11. Леоненков А. В. Нечеткое моделирование в среде MATLAB и fuzzyTECH. – СПб.: БХВ Петербург, 2005. 736 с.
12. Нечеткие множества в моделях управления и искусственного интеллекта/Под ред. Д. А. Поспелова.— М.: Наука. Гл. ред. физ.-мат. лит., 1986.— 312 с
13. М. В. Ayhana. Fuzzy AHP Approach for supplier selection problem: a case study in a gearmotor company. International Journal of Managing Value and Supply Chains (IJMVSC) Vol.4, No. 3, September 2013
14. H. Zhang. The Analysis of the Reasonable Structure of Water Conservancy Investment of Capital Construction in China by AHP Method. Water Resour Manage (2009) 23:1–18
15. Харари Ф. Теория графов. - М.: Мир, 1973.

ПРИЛОЖЕНИЕ

Листинг 5.1: Класс нечетких чисел

```
1  public class FuzzyNumber
2  {
3      public double a;
4      public double alpha;
5      public double beta;
6
7
8      public FuzzyNumber(double a1, double alpha1, double beta1)
9      {
10         a = a1;
11         alpha = alpha1;
12         beta = beta1;
13     }
14
15
16     public static FuzzyNumber operator +(FuzzyNumber f1,
17         FuzzyNumber f2) =>
18         new FuzzyNumber(f1.a + f2.a, f1.alpha + f2.alpha, f1.beta + f2.
19             beta);
20
21
22     public static FuzzyNumber operator -(FuzzyNumber f1,
23         FuzzyNumber f2) =>
24         new FuzzyNumber(f1.a - f2.a, f1.alpha + f2.beta, f1.beta + f2.
25             alpha);
26
27
28     public static FuzzyNumber operator *(FuzzyNumber f1,
29         FuzzyNumber f2) =>
30         new FuzzyNumber(f1.a * f2.a, f1.a * f2.alpha + f2.a * f1.alpha,
31             f1.a * f2.beta + f2.a * f1.beta);
32
33
34     public static FuzzyNumber operator /(FuzzyNumber f1,
35         FuzzyNumber f2) =>
36         new FuzzyNumber(f1.a / f2.a, (f1.a * f2.beta + f2.a * f1.alpha)
37             / Math.Pow(f2.a,2), (f1.a * f1.alpha + f2.a * f1.beta) /
```

```
Math.Pow(f2.a, 2));
```

27

```
28 public override string ToString() => $"({this.alpha} , {this.a}  
    , {this.beta})";
```

Листинг 5.2: Класс для анализа иерархий

```
1 partial class Program  
2 {  
3     public static bool CheckBasics(double[,] matrix)  
4     {  
5         List<int> leftGood = new List<int> { };  
6         List<int> rightGood = new List<int> { };  
7         List<int> leftBad = new List<int> { };  
8         List<int> rightBad = new List<int> { };  
9         bool Checker = false;  
10        int i1, j1;  
11        for (int i = 0; i < matrix.GetLength(0); i++)  
12        {  
13            for (int j = i+1; j < matrix.GetLength(0); j++)  
14            {  
15                if (matrix[i, j] != 0)  
16                {  
17                    i1 = leftGood.IndexOf(i);  
18                    j1 = rightGood.IndexOf(j);  
19                    if (i1 != -1)  
20                    {  
21                        if (leftBad.Contains(i)) { Checker = true; }  
22                        rightBad.Add(j);  
23                        rightBad.Add(rightGood[i1]);  
24                        leftGood.Add(i);  
25                    }  
26                    else { leftGood.Add(i); }  
27                    if (j1 != -1)  
28                    {  
29                        if (rightBad.Contains(j)) { Checker = true; }  
30                        leftBad.Add(i);  
31                        leftBad.Add(leftGood[j1]);  
32                        rightGood.Add(j);  
33                    }  
34                    else { rightGood.Add(j); }
```

```

35         }
36     }
37 }
38     return Checker;
39 }
40 public static void RecursiveFill(ref FuzzyNumber[,] matrix ,
    int index1, int index2, FuzzyNumber value)
41 {
42     List<int> NextLeftElem = new List<int> { };
43     List<int> NextRightElem = new List<int> { };
44     int k = LeftIndexes.IndexOf(index2, 0);
45     while (k != -1)
46     {
47         if (k != -1)
48             if (RightIndexes[k] != index1)
49                 { NextLeftElem.Add(RightIndexes[k]); LeftIndexes.
                    RemoveAt(k); RightIndexes.RemoveAt(k); }
50             k = LeftIndexes.IndexOf(index2, k);
51     }
52     k = RightIndexes.IndexOf(index2, 0);
53     while (k != -1)
54     {
55         if (k != -1)
56             if (LeftIndexes[k] != index1)
57                 { NextRightElem.Add(LeftIndexes[k]); LeftIndexes.
                    RemoveAt(k); RightIndexes.RemoveAt(k); }
58             k = RightIndexes.IndexOf(index2, k);
59     }
60     matrix[index2, array.GetLength(0)-1] = value;
61     if (NextRightElem.Count != 0)
62     {
63         foreach (int i in NextRightElem)
64         {
65             RecursiveFill(ref array, index2, i, matrix[i, index2]
                * value);
66         }
67     }
68     if (NextLeftElem.Count != 0)
69     {
70         foreach (int i in NextLeftElem)

```

```

71         {
72             RecursiveFill(ref array, index2, i, value / matrix[
                index2, i]);
73         }
74     }
75 }
76 }

```

Листинг 5.3: Простая реализация интегрирования

```

1  public delegate double Function(double x, FuzzyNumber y);
2  class Integrate
3  {
4      public static double integrate(Function func, double left,
        double right, FuzzyNumber f)
5      {
6          double l = 0;
7          double step = (right - left) / 2000;
8          for (double a = left; a <= right; a = a + step)
9          {
10             l = l + ((func(a, f) + func(a + step, f)) / 2) * (step);
11         }
12
13         return l;
14     }
15 }

```

Листинг 5.4: Выдержка из метода main

```

1  static void Main(string[] args)
2  {
3      int n = array2.GetLength(0);
4      for (int i = 0; i < n; i++)
5      {
6          for (int j = 0; j < n; j++)
7          {
8              array[i, j] = new FuzzyNumber(array2[i, j], array1[i, j], array3
                [i, j]);
9              if ((j > i) && (array2[i, j] != 0)) { LeftIndexes.Add(i);
                RightIndexes.Add(j); }
10         }
11     }

```

```

12 FuzzyNumber value = new FuzzyNumber(1, 1, 1);
13 RecursiveFill(ref array, n - 1, n - 1, value);
14 for (int i = 0; i < n; i++)
15 {
16     for (int j = 0; j < n; j++)
17     {
18         Console.Write(array[i, j].ToString());
19     }
20     Console.WriteLine();
21 }
22 double sum = 0;
23 double temp = 0;
24
25 List<double> final = new List<double> { };
26 for (int i = 0; i < n; i++)
27 {
28     temp = Integrate.integrate(testf1, -20, 60, array[i, n - 1]) /
29         Integrate.integrate(testf2, -20, 60, array[i, n - 1]);
30     final.Add(temp);
31     sum += temp;
32 }
33 double sum2 = 0;
34 for (int i = 0; i < n; i++)
35 {
36     Console.WriteLine(final[i] / sum);
37 }
38 Console.WriteLine();
39 Console.ReadKey();
40 }

```